ソフトウェア基礎論配布資料 (5)
単純型付 λ 計算

五十嵐 淳
京都大学 大学院情報学研究科知能情報学専攻
e-mail: igarashi@kuis.kyoto-u.ac.jp

平成 19 年 12 月 4 日

1 動機づけ

- 部分項として `1 - true` のようなものを含む項はエラーの発生を示していると考えられる.
- 静的に安全性 (エラーが発生しないこと) の保証をしたい. つまり, プログラム (項) を実行 (簡約) する前に, こういった演算子とその引数のミスマッチに起因するエラーの発生を検知したい. ただし, このような問題自体は決定不能なので, エラーが発生するプログラムは全て検出するが, エラーが発生しないプログラムも危険と判断されうる, という保守的な方法をとる.
- 型システム:

  > プログラム (項) を, その構文的情報のみを使って, その予想される実行結果 (簡約の正規形) の種類で分類するための仕組み

2 単純型付き λ 計算

λ 計算に対して考えられる最も単純な型システムを形式化したものを**単純型付 λ 計算**(*simply typed λ-calculus*) と呼ぶ. これに対して, 前回まで扱った体系を**型無し λ 計算**(*untyped λ-calculus*) と呼ぶ.

2.1 型

項の分類に用いる情報を**型**(*type*) と呼ぶ．型としてどんなものを用意するかで，受理できるプログラムの集合の大きさ，保証できる安全性 (すなわち，排除できるエラーの種類) が大きく異なる．ここでは，エラーとして，

- 関数でないものの適用 (`true 3` など)
- 整数でないものの減算・比較 (`4 - `$\lambda x.x$ や `false < 0` など)
- 真偽値でないものでの条件分岐 (`if 4 then ... else ...` など)

を考える．そのために型として，

- 整数に評価される項のための型 `int`
- 真偽値に評価される項のための型 `bool`
- 関数値 (λ 抽象) に評価される項のための型

を考える．さらに，関数の型は定義域 (の型) と値域 (の型) によって細かく分け，$T_1 \to T_2$ で「T_1 型の項を引数として T_2 型の項を返す関数」の型を表わす．例えば $\texttt{int} \to \texttt{int}$ は自然数上の関数の型であり，$(\texttt{int} \to \texttt{int}) \to \texttt{bool}$ は自然数上の関数を引数として真偽値を返す関数の型である．`int`, `bool` のような，型の原子とも言える型を**基底型**(*base type*) と呼ぶことがある． また，このように基底型と関数型だけからなる型 (の構造) を**単純型**(*simple type*) と呼ぶ.

2.2 型付け関係 (型判断)，型環境と型付け規則

項 t が何らかの型 T の分類にあてはまる時，$t \in T$ と書き，項 t の型が T である，とか，単に項が**型付け可能**(*typable*) という．例えば $\texttt{true} \in \texttt{bool}$ や $\texttt{0} \in \texttt{int}$ が成立すると考えられる．また，t_1 `-` t_2 のような部分式を持つ項に関しては,「t_1, t_2 が `int` 型ならば t_1 `-` t_2 は `int` 型である」というような規則が考えられる．このような項に型を与えるための規則を**型付け規則**(*typing rule*) と呼ぶ.

型付け規則は，型無し λ 計算における項の構成規則 (「環境 Γ の下で t は項である」という $\Gamma \vdash t \in \mathbf{Term}$ の導出規則) をさらに精密にし,「環境 Γ の下で t は T 型の項である」という意味の $\Gamma \vdash t \in T$ という判断を導くための規則と捉えることができる．この判断を特に**型付け判断**(*type judgment*) と呼ぶ．λ 計算では環境の中の宣言 x は「x は λ 項である」という意味であったが，単純型付 λ 計算では，宣言として $x \in T$ という「x は T 型の項である」という意味のものを考える．同時に定義も $x \in T = t$ という形のものを考える．このような型情報が付加された宣言列 (環境) を特に**型環境**(*type environment*) ということがある．(以前と同様に定義だけからなる環境を Δ で表す．)

型付け規則の中で鍵となるのは，変数参照，関数適用，関数抽象の規則である．変数参照に関しては宣言された型がそのまま項の型になる．

$$\frac{\Gamma \in \mathbf{Env}}{\Gamma, x \in T \vdash \#^0 x \in T} \qquad \text{(T-Var0)}$$

関数を構成するための λ 抽象項には期待通り関数型が与えられる．定義域・値域の型はどのように決まるのだろう．ここでは，定義域の型は，プログラマが宣言するものと考え，宣言と同様，項の文法にも型宣言を加え，λ 抽象項は $\lambda x{\in}T.t$ と記述する．現在の環境に局所的に $x \in T_1$ という宣言を加えた環境の下で $t \in T_2$ である，ということは x として，T_1 型のどんな λ 項が引数として渡されても，t は T_2 である，ということでもあるので，$\lambda x{\in}T.t$ 全体は T_1 から T_2 への関数として振舞うことができる．

$$\frac{\Gamma, x \in T_1 \vdash t \in T_2}{\Gamma \vdash \lambda x{\in}T_1.t \in T_1 \to T_2} \qquad \text{(T-Abs)}$$

関数適用項 $t_1\ t_2$ の場合，t_1 がいずれ $\lambda x{\in}T.t_0$ という形になることを要請したいので，t_1 の型が関数型であることが必要になる．さらに，関数呼び出し後の定義域の型が引数 t_2 の型と合致している必要がある．

$$\frac{\Gamma \vdash t_1 \in T_1 \to T_2 \qquad \Gamma \vdash t_2 \in T_1}{\Gamma \vdash t_1\ t_2 \in T_2} \qquad \text{(T-App)}$$

2.3 型安全性

型付け可能な項に関しては，そもそもの動機づけで触れた，`1 - true` を実行するといった,「ある種のエラー」が発生しない，という性質が満たされていてほしい．このような型システムが提供する安全性を**型安全性**(*type safety*) という．型安全性は

- $\Gamma \vdash t \in T$ かつ $\Gamma \vdash t \longrightarrow t'$ ならば，$\Gamma \vdash t' \in T$ である．
- $\Delta \vdash t \in T$ かつ t が値でなければ，ある項 t' が存在して $\Delta \vdash t \longrightarrow t'$ である．

という，それぞれ Type Preservation, Progress と呼ばれる，ふたつの性質を組合せることで示される．つまり，型付け可能なプログラム (自由変数のない項) は `1 - true` のような，値でもないのに簡約しようのない項に辿りつくことがないことが示される．

ここでは，非決定的な簡約をプログラムの実行として定式化を行うが，評価関係を導入して同様な議論を行うことができる．算術式の体系で触れたような，評価関係に対応する決定的な簡約を導入すれば，非決定的な簡約に関する結果を適用することができる．

$$\frac{}{\bullet \in \mathbf{Env}} \quad \text{(Env-Empty)}$$

$$\frac{\Gamma \in \mathbf{Env}}{\Gamma, x \in T \in \mathbf{Env}} \quad \text{(Env-Decl)}$$

$$\frac{\Gamma \in \mathbf{Env} \qquad \Gamma \vdash t \in T}{\Gamma, x \in T = t \in \mathbf{Env}} \quad \text{(Env-Defn)}$$

$$\frac{\Gamma \in \mathbf{Env}}{\Gamma, x \in T \vdash \#^0 x \in T} \quad \text{(T-Var0)}$$

$$\frac{\Gamma \in \mathbf{Env} \qquad \Gamma \vdash t \in T}{\Gamma, x \in T = t \vdash \#^0 x \in T} \quad \text{(T-Var1)}$$

$$\frac{\Gamma \vdash \#^i x \in T}{\Gamma, x \in T' \vdash \#^{i+1} x \in T} \quad \text{(T-Var2)}$$

$$\frac{\Gamma \vdash \#^i x \in T \qquad \Gamma \vdash t \in T'}{\Gamma, x \in T' = t \vdash \#^{i+1} x \in T} \quad \text{(T-Var3)}$$

$$\frac{\Gamma \vdash \#^i x \in T \qquad (x \not\equiv y)}{\Gamma, y \in T' \vdash \#^i x \in T} \quad \text{(T-Var4)}$$

$$\frac{\Gamma \vdash \#^i x \in T \qquad \Gamma \vdash t \in T' \qquad (x \not\equiv y)}{\Gamma, y \in T' = t \vdash \#^i x \in T} \quad \text{(T-Var5)}$$

$$\frac{\Gamma \vdash t_1 \in T_1 \to T_2 \qquad \Gamma \vdash t_2 \in T_1}{\Gamma \vdash t_1\ t_2 \in T_2} \quad \text{(T-App)}$$

$$\frac{\Gamma, x \in T_1 \vdash t \in T_2}{\Gamma \vdash \lambda x \in T_1.t \in T_1 \to T_2} \quad \text{(T-Abs)}$$

$$\frac{\Gamma \in \mathbf{Env} \qquad (n \text{ is an integer})}{\Gamma \vdash n \in \texttt{int}} \quad \text{(T-Num)}$$

$$\frac{\Gamma \vdash t_1 \in \texttt{int} \qquad \Gamma \vdash t_2 \in \texttt{int}}{\Gamma \vdash t_1\ \texttt{-}\ t_2 \in \texttt{int}} \quad \text{(T-Minus)}$$

$$\frac{\Gamma \in \mathbf{Env}}{\Gamma \vdash \texttt{true} \in \texttt{bool}} \quad \text{(T-True)}$$

$$\frac{\Gamma \in \mathbf{Env}}{\Gamma \vdash \texttt{false} \in \texttt{bool}} \quad \text{(T-False)}$$

$$\frac{\Gamma \vdash t_1 \in \texttt{int} \qquad \Gamma \vdash t_2 \in \texttt{int}}{\Gamma \vdash t_1\ \texttt{>}\ t_2 \in \texttt{bool}} \quad \text{(T-Gt)}$$

$$\frac{\Gamma \vdash t_1 \in \texttt{bool} \qquad \Gamma \vdash t_2 \in T \qquad \Gamma \vdash t_3 \in T}{\Gamma \vdash \texttt{if}\ t_1\ \texttt{then}\ t_2\ \texttt{else}\ t_3 \in T} \quad \text{(T-If)}$$

$$\frac{\Gamma, f \in T \to T, x \in T \vdash t \in T}{\Gamma \vdash \texttt{rec}\ f.\lambda x.t \in T \to T} \quad \text{(T-Fix)}$$

図 1: 単純型付 λ 項の定義

3 形式的定義

3.1 定義: 型 T は以下の文法で定義される.

$$T ::= \texttt{int} \mid \texttt{bool} \mid T \to T$$

□

3.2 定義: 判断 $\Gamma \in \mathbf{Env}$ と型判断 $\Gamma \vdash t \in T$ は図 1 の規則で定義される. □

3.3 定義: 簡約関係の判断 $\Gamma \vdash t \longrightarrow t'$ は以下の規則で定義される. ($\Gamma \vdash t[\#^i x] \Rightarrow t'$ の定義は省略している.) □

$$\frac{}{\Gamma, x \in T = t \vdash x \longrightarrow t\uparrow_x} \quad \text{(R-Def)}$$

$$\frac{\Gamma \vdash \mathtt{\#}^i x \longrightarrow t}{\Gamma, y \in T \vdash \mathtt{\#}^i x \uparrow_y \longrightarrow t \uparrow_y} \quad \text{(R-Shift1)}$$

$$\frac{\Gamma \vdash \mathtt{\#}^i x \longrightarrow t}{\Gamma, y \in T' = t' \vdash \mathtt{\#}^i x \uparrow_y \longrightarrow t \uparrow_y} \quad \text{(R-Shift2)}$$

$$\frac{\Gamma \vdash t_2 \in \mathbf{VTerm} \qquad \Gamma, x \in T_2 = t_2 \vdash t_1[\mathtt{\#}^0 x] \Rightarrow t'}{\Gamma \vdash (\lambda x \in T_2.t_1)\ t_2 \longrightarrow t' \downarrow_x} \quad \text{(R-BetaV)}$$

$$\frac{(n_3 = n_1 - n_2)}{\Gamma \vdash n_1\ \mathtt{-}\ n_2 \longrightarrow n_3} \quad \text{(R-Minus)}$$

$$\frac{(n_1 > n_2)}{\Gamma \vdash n_1\ \mathtt{>}\ n_2 \longrightarrow \mathtt{true}} \quad \text{(R-GtT)}$$

$$\frac{(n_1 \leq n_2)}{\Gamma \vdash n_1\ \mathtt{>}\ n_2 \longrightarrow \mathtt{false}} \quad \text{(R-GtF)}$$

$$\frac{}{\Gamma \vdash \mathtt{if\ true\ then}\ t_1\ \mathtt{else}\ t_2 \longrightarrow t_1} \quad \text{(R-IfT)}$$

$$\frac{}{\Gamma \vdash \mathtt{if\ false\ then}\ t_1\ \mathtt{else}\ t_2 \longrightarrow t_2} \quad \text{(R-IfF)}$$

$$\frac{\Gamma, f \in T \to T = \mathtt{rec}\ f.\lambda x \in T.t \vdash \lambda x \in T.t[\mathtt{\#}^0 f] \Rightarrow t'}{\Gamma \vdash \mathtt{rec}\ f.\lambda x \in T.t \longrightarrow t' \downarrow_f} \quad \text{(R-Rec)}$$

$$\frac{\Gamma \vdash t_1 \longrightarrow t_1'}{\Gamma \vdash t_1\ t_2 \longrightarrow t_1'\ t_2} \quad \text{(R-AppL)}$$

$$\frac{\Gamma \vdash t_2 \longrightarrow t_2'}{\Gamma \vdash t_1\ t_2 \longrightarrow t_1\ t_2'} \quad \text{(R-AppR)}$$

$$\frac{\Gamma, x \in T \vdash t \longrightarrow t'}{\Gamma \vdash \lambda x{\in}T.t \longrightarrow \lambda x{\in}T.t'} \quad \text{(R-Abs)}$$

$$\frac{\Gamma \vdash t_1 \longrightarrow t_1'}{\Gamma \vdash t_1\ \mathtt{-}\ t_2 \longrightarrow t_1'\ \mathtt{-}\ t_2} \quad \text{(R-MinusL)}$$

$$\frac{\Gamma \vdash t_2 \longrightarrow t_2'}{\Gamma \vdash t_1\ \mathtt{-}\ t_2 \longrightarrow t_1\ \mathtt{-}\ t_2'} \quad \text{(R-MinusR)}$$

$$\frac{\Gamma \vdash t_1 \longrightarrow t_1'}{\Gamma \vdash t_1\ \mathtt{<}\ t_2 \longrightarrow t_1'\ \mathtt{<}\ t_2} \quad \text{(R-GtL)}$$

$$\frac{\Gamma \vdash t_2 \longrightarrow t_2'}{\Gamma \vdash t_1\ \mathtt{<}\ t_2 \longrightarrow t_1\ \mathtt{<}\ t_2'} \quad \text{(R-GtR)}$$

$$\frac{\Gamma \vdash t_1 \longrightarrow t_1'}{\Gamma \vdash \mathtt{if}\ t_1\ \mathtt{then}\ t_2\ \mathtt{else}\ t_3 \longrightarrow \mathtt{if}\ t_1'\ \mathtt{then}\ t_2\ \mathtt{else}\ t_3} \quad \text{(R-If1)}$$

$$\frac{\Gamma \vdash t_2 \longrightarrow t_2'}{\Gamma \vdash \mathtt{if}\ t_1\ \mathtt{then}\ t_2\ \mathtt{else}\ t_3 \longrightarrow \mathtt{if}\ t_1\ \mathtt{then}\ t_2'\ \mathtt{else}\ t_3} \quad \text{(R-If2)}$$

$$\frac{\Gamma \vdash t_3 \longrightarrow t_3'}{\Gamma \vdash \mathtt{if}\ t_1\ \mathtt{then}\ t_2\ \mathtt{else}\ t_3 \longrightarrow \mathtt{if}\ t_1\ \mathtt{then}\ t_2\ \mathtt{else}\ t_3'} \quad \text{(R-If3)}$$

$$\frac{\Gamma \vdash t_0 \longrightarrow t_0'}{\Gamma \vdash \mathtt{fix}\ t_0 \longrightarrow \mathtt{fix}\ t_0'} \quad \text{(R-Fix0)}$$

図 2: 単純型付 λ 計算: (値呼び) 簡約関係

4 型検査アルゴリズム

型付け規則を下から上に読むと，型検査をするアルゴリズムが得られる．環境Γ，項(らしきもの)tを入力として，$\Gamma \vdash t \in T$ なるT (あれば) を返す，アルゴリズム$TC(\Gamma, t)$は以下のように記述できる．

$$
\begin{array}{ll}
TC((\Gamma, x \in T), \#^0 x) = & \textbf{if } TC_{env}(\Gamma) \textbf{ then } T \textbf{ else } \mathit{fail} \\
TC((\Gamma, x \in T = t), \#^0 x) = & \textbf{if } TC_{env}(\Gamma) \textbf{ then } T \textbf{ else } \mathit{fail} \\
TC((\Gamma, x \in T'), \#^{i+1} x) = & TC(\Gamma, \#^i x) \\
\quad \vdots & \\
TC(\Gamma, t_1\ t_2) = & \textbf{let } T_{11} \to T_{12} = TC(\Gamma, t_1) \textbf{ in} \\
 & \textbf{let } T_2 = TC(\Gamma, t_2) \textbf{ in} \\
 & \textbf{if } T_{11} = T_2 \textbf{ then } T_{12} \textbf{ else } \mathit{fail} \\
TC(\Gamma, \lambda x{\in}T_1.t) = & \textbf{let } T_2 = TC((\Gamma, x \in T_1), t) \textbf{ in } T_1 \to T_2 \\
TC(\Gamma, n) = & \texttt{int} \\
TC(\Gamma, t_1\ \texttt{-}\ t_2) = & \textbf{let } T_1 = TC(\Gamma, t_1) \textbf{ in} \\
 & \textbf{let } T_2 = TC(\Gamma, t_2) \textbf{ in} \\
 & \textbf{if } T_1 = \texttt{int} \,\&\, T_2 = \texttt{int} \textbf{ then } \texttt{int} \textbf{ else } \mathit{fail} \\
\quad \vdots &
\end{array}
$$

5 諸性質

5.1 定理 [Uniqueness of Typing]: $\Gamma \vdash t \in T$ かつ $\Gamma \vdash t \in T'$ ならば$T \equiv T'$ である．

5.2 定理 [Type Preservation]: $\Gamma \vdash t \in T$ かつ $\Gamma \vdash t \longrightarrow t'$ ならば，$\Gamma \vdash t' \in T$ である．

5.3 定理 [Progress]: $\Delta \vdash t \in T$ ならば t は $\texttt{true}, \texttt{false}, n, \lambda x{\in}T.t_0$ であるか，ある項 t' が存在して $\Gamma \vdash t \longrightarrow t'$ である．

5.4 定理 [Strong Normalization]: $\Gamma \vdash t \in T$ かつ Γ, t が rec を含まないならば$\Gamma \vdash t \longrightarrow t_1 \longrightarrow \cdots \longrightarrow t_n \longrightarrow \cdots$ なる無限列は存在しない．

6 Type Preservation と Progress の略証

6.1 補題 [Inversion]:
1. もし $\Gamma \vdash \lambda x{\in}T_1.t_0 \in T$ ならば，ある T_2 が存在して $T \equiv T_1 \to T_2$ かつ $\Gamma, x \in T_1 \vdash t_0 \in T_2$ が成立する．
2. もし $\Gamma \vdash t_1\ t_2 \in T$ ならば，T_1 が存在して$\Gamma \vdash t_1 \in T_1 \to T$ かつ $\Gamma \vdash t_2 \in T_1$ が成立する．

3. もし $\Gamma \vdash \texttt{if}\ t_1\ \texttt{then}\ t_2\ \texttt{else}\ t_3 \in T$ ならば，$\Gamma \vdash t_1 \in \texttt{bool}$ かつ $\Gamma \vdash t_2 \in T$ かつ $\Gamma \vdash t_3 \in T$

などなど．

Proof: (変数参照項を除いて) 項について適用できる型付け規則がひとつなので，型付け規則を逆 (下から上) に読むことによって明らか． □

6.2 補題 [Substitution Preserves Typing]: $\Gamma, x \in T' \vdash t \in T$ かつ $\Gamma \vdash t' \in T'$ かつ $\Gamma, x \in T' = t' \vdash t[x] \Rightarrow t''$ ならば，$\Gamma, x \in T' \vdash t'' \in T$ が成立する．

Proof: 上の文言より，少し強い性質である，

> $\Gamma, x \in T', \Gamma' \vdash t \in T$ かつ $\Gamma \vdash t' \in T'$ かつ $\Gamma, x \in T' = t', \Gamma' \vdash t[\#^i x] \Rightarrow t''$ かつ Γ' 中には x の宣言は i 個出現するならば，$\Gamma, x \in T', \Gamma' \vdash t'' \in T$ が成立する．

を $\Gamma, x \in T' = t', \Gamma' \vdash t[\#^i x] \Rightarrow t''$ の導出に関する帰納法によって示す．途中，$\Gamma \vdash t \in T$ ならば $\Gamma, x \in T' \vdash t\uparrow_x \in T$ という性質を使う．これが示されれば，$\Gamma' \equiv \bullet$ の場合を考えればよい． □

Proof of Type Preservation: $\Gamma \vdash t \longrightarrow t'$ の導出に関する帰納法で証明する．帰納法の base に相当するのは R-Beta など前提に $\Gamma \vdash t \longrightarrow t'$ の形が現れない規則で $\Gamma \vdash t \longrightarrow t'$ が導出された場合である．

- 規則 R-Beta の場合，...
- 規則 R-Rec の場合，...
- 規則 R-Minus の場合，...

などなど．

帰納法のステップに相当するのは，前提のある規則で $\Gamma \vdash t \longrightarrow t'$ が導出された場合である．この時，前提として成立している $\Gamma \vdash t_0 \longrightarrow t_0'$ に関しては Type Preservation が成立しているとしてよい (帰納法の仮定)．

- 規則 R-AppL の場合，ある t_1, t_2, t_1' が存在して $t \equiv t_1\ t_2$ かつ $t' \equiv t_1'\ t_2$ かつ $\Gamma \vdash t_1 \longrightarrow t_1'$ が成立する．Inversion より，ある T_1 が存在して $\Gamma \vdash t_1 \in T_1 \to T$ かつ $\Gamma \vdash t_2 \in T_1$ である．帰納法の仮定より $\Gamma \vdash t_1' \in T_1 \to T$ が成立する．T-App より $\Gamma \vdash t_1'\ t_2 \in T$ が導出できる．

などなど． □

6.3 補題 [Canonical Forms]: $\Gamma \equiv x_1 \in T_1 = t_1, \ldots, x_n \in T_n = t_n$ かつ $\Gamma \vdash t \in T$ かつ t は正規形であるとする．

1. もし $T \equiv T_1' \to T_2'$ ならば，ある x, t_0 が 存在して $t \equiv \lambda x{\in}T_1'.t_0$ が成立する．
2. もし $T \equiv \texttt{int}$ ならば， t は整数である．
3. もし $T \equiv \texttt{bool}$ ならば， $t \equiv \texttt{true}$ または $t \equiv \texttt{false}$ である．

Proof of Progress: 型付け関係 $\Gamma \vdash t \in T$ に関する帰納法で証明する．t の形で場合分けを行なう．

- $t \equiv \texttt{\#}^n x$ の場合は，Γ の形から $\Gamma \vdash \texttt{\#}^n x \longrightarrow t'$ なる t' が存在する．
- $t \equiv \lambda x{\in}T.t_0$ の場合は，これで OK.
- $t \equiv t_1\ t_2$ の場合，Inversion より，ある T_1 が存在して $\Gamma \vdash t_1 \in T_1 \to T$ かつ $\Gamma \vdash t_2 \in T_1$ が成立する．帰納法の仮定より t_1 が値であるか，ある t_1' が存在して $\Gamma \vdash t_1 \longrightarrow t_1'$ である．前者の場合，Canonical Forms より $t_1 \equiv \lambda x{\in}T_1.t_0$ であるので，R-Beta より，ある t' が存在して $\Gamma \vdash t_1\ t_2 \longrightarrow t'$ である．一方，後者の場合，R-AppL より $\Gamma \vdash t_1\ t_2 \longrightarrow t_1'\ t_2$ を導出することができ，結局，どちらの場合でも $\Gamma \vdash t \longrightarrow t'$ なる t' が存在する．

などなど． □